Asset-based Documentation in an Agile Development Environment

This article profiles asset-based documentation in an agile development environment. To illustrate this idea I've used the example of the agile development environment in place at Flashline and the documentation model I developed to work in that environment, based on needed documentation deliverables.

Asset-based documentation is really nothing more than documentation built on the same principles of asset-based software engineering. But first -- what is an agile development environment?

Agile Development – in a Nutshell

Agile development, as practiced at Flashline, is a set of techniques based primarily on the concept of a rugby scrum, as described in *Agile Software Development with Scrum (***ISBN:** 0130676349*)* and *Adaptive Software Development (***ISBN:** 0932633404 *)*. Keys to the concept of agility are the notions of time-specific iterative development cycles and self-organizing teams. Running parallel to those notions is a close interaction between product management and development to manage the priorities and delivery of product enhancements.

Think about the "old-fashioned" way of developing software. Everyone in the organization gets involved in the "vision" of a new product. There is a high-level, top-down approach to the product where requirements are refined for development. After the product folks, the marketing folks, the sales folks, and the development folks have all chewed on the vision, everyone separates to perform their part of the process. Months can go by, and marketing and sales campaigns are cogitated, enhanced, and brought to life. Meanwhile, developers have variously and severally set about writing code that they believe fulfills the vision of the product that they are to deliver.

Finally, the alpha phase arrives, with delivery scheduled for internal customers! But wait – this isn't what the marketing, sales and product folks envisioned at all – or not exactly. This is close, but it's lacking functionality that was promised to the field, or has interface issues or…you get the picture! A great deal of time spent by a number of different people has resulted in something – but is it the something that fulfills the vision and customer requirements? And does it work?

In agile development, the concept of deliverables at specific milestones seeks to keep the development cycle more flexible, while still providing checkpoints -- or reality checks, if you will -- to ensure that development is actually on track with what product management and marketing envision.

Following the rugby model, development teams are known as scrum teams, and the time-specific period is a sprint. In Flashline's case, a sprint is normally four weeks. The team sprints to achieve goals that they set themselves, with input from product management. Implementing agile techniques enables substantial flexibility to meet changing requirements, while also enabling the delivery of a complete, release-ready product on a timely basis.

As you might expect, getting buy-in from team members, and changing "old development process" habits, may be a challenge. Agile development practices, however, enable a degree of autonomy to the development team, which further enables their voice in the process, as well as in problem-solving, and strategizing.

In a nutshell, then, agile development enables a focused, directed cycle with a verifiable deliverable at the end of each sprint. While a sprint deliverable may not be a salable item, the process yields the modules or components to build a tested, viable application at the end of a larger cycle.

How does Asset-based Documentation work in an Agile Environment?

Beautifully! That's the short answer, of course, but true. In my situation I'm able to choose how to produce the documentation that accompanies our product to market. While this may be the moral equivalent of being the Empress of Everything, it's really one more R & D member trying to produce the highest quality product in the most efficient way possible. For Flashline, that translates to using existing assets whenever possible. In producing documentation, asset-based documentation enables me to make changes in a file once, then build the deliverables without the anxiety of wondering if a change was made everyplace it needed to be.

In the same way that modular software components are produced during a sprint, documentation is produced in a modular fashion. For instance, user information is produced in parallel as application functions are developed; system administration information is also produced in parallel as the functionality of the system emerges. This doesn't mean that a completed document exists at the end of a sprint – rather the functions inherent in the sprint are documented as they are built and later become part of a whole document or help system.

What's Involved in Producing Asset-based Documentation?

In some situations you may be faced with outdated, sketchy, or inconsistent documentation. Add a number of diverse authors with differing interests and skill sets, and effective, usable documentation is a challenge!

In our case, at Flashline, producing documentation included several steps. The first was finding the most recent version of the documentation. After reviewing this documentation, a determination was made about the changes or additions necessary to bring it up to speed, and how it should be used and stored. The last part of the process was to “build” or “package” the documentation shipped with our products

First Steps

The first step is to find the most recent version of the existing documentation. After you identify and review the most recent version of the documentation it may be necessary to separate it into information for end users and information for admin types, either at this stage or after revision.

In my case, the documentation was originally done in FrameMaker, from which help was generated. I made the decision to use Microsoft Word, because that was the standard in the office, and doing so allowed other people’s documentation to be melded into what I was doing much more easily. Accordingly, all documentation was “pulled apart” and put into separate files that could be properly termed “info bites.” Each bite, or topic, covers only one thing – for instance, a procedure for end users, introductory material, or a procedure for admin people.

Interim Steps

After re-writing and editing existing modules, and writing new modules, modules were separated by user function and stored electronically in folders that contained only files for end-users or for admins. Installation documentation also has its own folder. Modular files used for all functions went in a common folder where they were readily accessible.

At the end…

The packaging, or formation of final documentation, is the last phase of the asset-based documentation process. Using the modular parts in each folder, a book is built in Word by inserting files into a larger file. The “book” file, when finished, has a TOC and index, and is then converted to PDF using a Word plug-in. Help files were built using the same files used for a book. Originally, the Word modules were turned into help files in Front Page. Today, those files are processed in Homesite™ as XHTML files, which make them more compatible with our application. The third format used is compiled help files, using Microsoft Help Workshop. And finally, the book file is used to create Microsoft Reader files that can be downloaded successfully from a Web site and is an easy-reading alternative to PDFs.

Asset-based documentation, like asset-based software, follows Flashline Registry’s™ model of finding and utilizing existing assets to build a new product.